C# Cumulative Part 3

By: Christine Bittle

This cumulative project involves building a Minimum Viable Product (MVP) on the Teachers table of the provided School Database using ASP.NET Core Web API and MVC. Part 3 is the Update functionality

MVP: Create, Read, Update, Delete on the Teachers Table.

Type	Description	File
-	A class which connects to your MySQL database	`/Models /SchoolDbContext.cs`
Controller	A WebAPI Controller which allows you to Update teachers	`/Controllers /TeacherAPIController.cs`
Controller	An MVC Controller which allows you to route to dynamic pages	`/Controllers /TeacherPageController.cs`
Model	A Model which allows you to represent information about a teacher	`/Models /Teacher.cs`
View	A View which uses server rendering to display a page that allows for a user to Update a teacher	`/Views /Teacher /Edit.cshtml`

How to submit

1. Use Visual Studio / git to push your work to a remote repository (Parts 1, 2, and 3 can be on the same repository)
2. Verify the repository:
    a. contains the work you wish to submit (i.e. the files are there)
    b. is public (if it is set to private, change it to public!)
3. Include repository github link as part of your assignment submission (Do not share the link or your work with anyone else)
4. Include evidence of your testing as a PDF with screenshots of your cURL commands and web pages

(Quantitative: 8 Marks): Implement the required MVP using ASP.NET Core:

- (4 Marks) An API which Updates a Teacher
- (4 Marks) A web page that allows a user to enter updated Teacher information

(Qualitative: 8 Marks): Document your work with (a) descriptive:

- (2 Marks) Summary blocks for your API methods
- (2 Marks) Teacher Model Properties (Teacher.cs)
- (2 Marks) Variable names
- (2 Marks) Project .readme

(Testing: 8 Marks): Include evidence of the following testing:

- (4 Marks) Your API that updates a Teacher using the HTTP PUT method
- (4 Marks) Your web page that allows a user to enter updated Teacher information

(Initiative: 8 Marks): Earn **up to 8** Initiative Marks by improving on MVP

- (2 Marks) (Server) Error Handling on Update when trying to update a teacher that does not exist
- (2 Marks) (Client) Error Handling on Update when the Teacher Name is empty
- (2 Marks) (Client) Error Handling on Update when the Teacher Hire Date is in the future
- (2 Marks) (Client) Error Handling on Update when the Salary is less than 0
- (2 Marks) (Server) Error Handling on Update when the Teacher Name is empty
- (2 Marks) (Server) Error Handling on Update when the Teacher Hire Date is in the future
- (2 Marks) (Server) Error Handling on Update when the Salary is less than 0
- (2 Marks) Update Functionality Students
- (2 Marks) Update Functionality Courses
- (2 Marks) Use JS and AJAX to Update a Teacher
- (2 Marks) Attempt a previously **un-attempted** initiative from C1 or C2 (Make sure to document which one you've attempted in your C3 submission to receive this grade!)

RUBRIC

	0 Marks	1 Mark	2 Marks
Quantitative	Functionality not implemented	Functionality partially implemented / implemented with room for improvements	Functionality implemented, no improvements required
Qualitative	Documentation not included	Documentation partially included / included with room for improvements	Documentation included, no improvements required
Testing	Testing not included	Testing partially included / included with room for improvements	Testing included, no improvements required
Initiative	No initiative elements attempted	Initiative elements partially implemented / implemented with room for improvements	Initiative included, no room for improvements